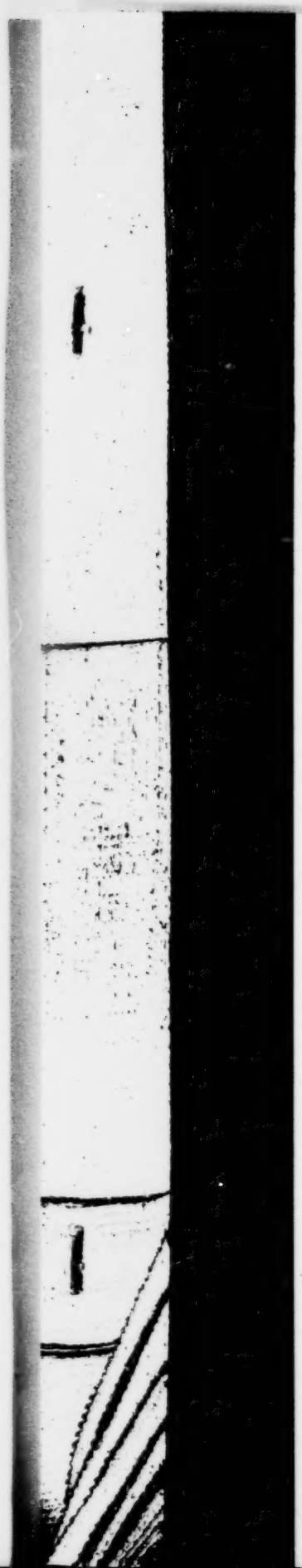

UNLIMITED
UNCLASSIFIED

ILLIMITÉE
NON CLASSIFIÉE

Technical Report

Rapport technique

1988 / 04

TR-SYS-017
NRC NO. 28863

KNOWLEDGE AND CONSTRAINT BASED SCHEDULING (KCBS) USING OPS5

Joseph Neelamkavil and Louis G. Birta

Division of
Mechanical Engineering

Division de
génie mécanique

National Research Council Canada
Conseil national de recherches Canada

Canada

89-00556/

DIVISION OF MECHANICAL ENGINEERING PUBLICATIONS

DM **(Division of Mechanical Engineering Report)**
Scientific and technical information considered important, complete and a lasting contribution to existing knowledge.

TR **(Technical Report)**
Information less broad in scope but a substantial contribution to existing knowledge.

CTR **(Controlled/Classified Technical Report)**
A Technical Report with controlled distribution for national security, proprietary or other reasons.

LM **(Laboratory Memorandum)**
Preliminary or exploratory information with controlled distribution.

CAT **(Calibration Analysis and Test Report)**
Information on minor laboratory projects or services.

PUBLICATIONS DE LA DIVISION DE GÉNIE MÉCANIQUE

DM **(Rapport de la Division de génie mécanique)**
Informations scientifiques et techniques jugées importantes, complètes et susceptibles de contribuer de façon durable à l'avancement des connaissances courantes.

TR **(Rapport technique)**
Informations de moindre importance, mais pouvant contribuer substantiellement à l'avancement des connaissances actuelles.

CTR **(Rapport technique à diffusion contrôlée/classifiée)**
Rapport technique à diffusion contrôlée pour des raisons de sécurité nationale, de propriété intellectuelle et autres.

LM **(Mémoire de laboratoire)**
Informations préliminaires ou de nature exploratoire à diffusion contrôlée.

CAT **(Rapport d'étalonnage d'analyse et d'essai)**
Informations sur de petits projets ou des services de laboratoire.

UNLIMITED
UNCLASSIFIED

ILLIMITÉE
NON CLASSIFIÉE

KNOWLEDGE AND CONSTRAINT BASED SCHEDULING (KCBS) USING OPS5

ORDONNANCEMENT À BASE DE CONNAISSANCES ET DE CONTRAINTES (OBCC) À L'AIDE DU OPS5

Joseph Neelamkavil* and Louis G. Birta**

*National Research Council Canada
**University of Ottava

Technical Report

Rapport technique

1988/04

TR-SYS-017
NRC No. 28863

V. Thomson, Head/Chef
Systems Laboratory
Laboratoire des systèmes

J. Ploeg
Director/
Directeur

ACKNOWLEDGMENTS

The scheduling system described in this report has been significantly influenced by the insights obtained during discussions with John Buckley, Udo Graefe, Izhar Mufti, and Vince Thomson of the Systems Laboratory. We are also grateful to Roger Fielding of ALCAN International who kindly introduced us to the heat treatment furnace problem referred to in this report.

ABSTRACT

This report describes a knowledge-based scheduling system and associated scheduling methodology, developed using the OPS5 expert system shell and FORTRAN interface routines. It is an experimental scheduling system created for production scheduling, which uses artificial intelligence concepts. At present, it is used for the scheduling of batch machine operations (e.g., heat treatment machines) where a batch consists of a number of parts having similar or dissimilar geometries but requiring identical processing (or operation), possibly with different due dates. The system can be conceptually viewed as consisting of two main modules: (a) an optimization module whose function is to create an optimum batch from a group of parts requiring identical processing and (b) a scheduling heuristic for the selection of the batch to be scheduled from among the various optimum batches. Optimization is performed by searching for the batch that maximizes the sum of job utilities (job importance) subject to various constraints; scheduling is done by striving to accommodate all resource constraints and other corporate manufacturing goals. If constraints such as due dates are not met, the system will attempt to satisfy these constraints by scheduling overtime on some appropriate machine.

RÉSUMÉ

Le présent rapport décrit un système d'ordonnancement à base de connaissances et la méthode d'ordonnancement rattachée, qu'on a mis au point à l'aide du système expert vide OPS5 et de programmes d'interface FORTRAN. Il s'agit d'un système expérimental créé en vue de l'ordonnancement de la production, basé sur des concepts d'intelligence artificielle. À l'heure actuelle, on l'utilise pour la planification du traitement de lots sur des machines (par exemple, sur des machines de traitement thermique), un lot consistant en un certain nombre de pièces de formes similiares ou différentes nécessitant un traitement (ou une opération) identifique, et pouvant avoir des échéances différentes. Le système peut être considéré essentiellement comme formé de deux modules principaux: (a) un module d'optimisation dont le rôle est de créer un lot optimum à partir d'un groupe de pièces nécessitant un traitement identique et (b) un module heuristique d'ordonnancement pour la sélection, parmi les différents lots optimums, du lot dont on veut faire l'ordonnancement. L'optimisation consiste à chercher le lot qui maximise la somme des priorités des travaux (importance des travaux) en fonction de différentes contraintes; l'ordonnancement consiste à essayer de tenir compte de toutes les contraintes sur le plan des ressources et des autres objectifs de fabrication de l'entreprise. Lorsque certaines contraintes, par exemple les échéances, ne sont pas respectées, le système assaie de les respecter en planifiant du travail supplémentaire sur une machine appropriée.

CONTENTS

		Page
	ACKNOWLEDGMENTS	(ii)
	ABSTRACT/RÉSUMÉ	(iii)
	LIST OF ILLUSTRATIONS	(v)
	LIST OF TABLES	(v)
1.0	INTRODUCTION	1
2.0	CONSTRAINT BASED SCHEDULING	1
3.0	KNOWLEDGE AND CONSTRAINT BASED SCHEDULING	1
4.0	MODEL DEVELOPMENT	3
	4.1 Model Parameters	3
	4.2 Model Representation	4
5.0	BATCH MACHINE SCHEDULING	6
	5.1 Batch Machine Problem Statement	7
	5.2 Batch Machine Constraints	7
	5.3 Batch Machine Solution Strategy	8
6.0	PROGRAM ORGANIZATION AND DEVELOPMENT	16
	6.1 The OPS5 Module	16
	6.2 The FORTRAN Module	17
7.0	TEST PROBLEM DESCRIPTION	18
8.0	DISCUSSION	21
9.0	CONCLUSIONS	21
10.0	REFERENCES	26
	DOCUMENTATION PAGE	

CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

Figure		Page
1	Structure of a Manufacturing Operation Model	5
2	Time Slot Diagram	9
3	Flowchart - Batch Machine Solution Procedure	15
4	Gantt Chart - Initial	20
5	Gantt Chart - With Overtime	22
6	Gantt Chart - Without Overtime	23

LIST OF TABLES

Table		
1	Customer Order Data	19
2	Schedule Comparison - Overtime vs No overtime	24
3	Lateness Comparison - Overtime vs No overtime	25

KNOWLEDGE AND CONSTRAINT BASED SCHEDULING (KCBS) USING OPS5

1.0 INTRODUCTION

In 1986, the Systems Laboratory, in collaboration with ICAM Technologies of Montreal, developed a job-shop scheduling system called FAMS [1]. At that time, it was realized that the scheduling of a batch machine (a machine in which a number of parts is processed together, e.g., heat treatment machine) should be explored in terms of other perspectives. The software and the heuristics given in this study, briefly reported elsewhere [2], were developed with that intent. However, the overall objective is the development of a comprehensive method for production scheduling which is able to take into account a wide variety of different conditions and constraints that affect schedule feasibility and quality.

2.0 CONSTRAINT BASED SCHEDULING

The constraint based scheduling problem is concerned with allocating a set of tasks to a set of resources, subject to a set of constraints. It can be characterized as having a large search space, which may be reduced through the examination and application of the relevant constraints. These constraints can take a variety of forms and can potentially involve all of the problem parameters. These include precedence constraints (some operations and/or jobs must precede others), physical constraints (a particular machine can process only parts whose length is less than 10), due date constraints (the final operation associated with the manufacturing of a product must be completed prior to a prescribed time), capacity constraints (the number of work hours per week can not exceed the plant capacity), and manufacturing goals (e.g., work-in-process inventory of the plant should be minimized and setup times must be reduced).

The constraint based scheduling approach is relatively recent; some of the relevant works are reported in references 3 to 25. Generally, the constraints can be separated into two broad categories that are normally referred to as hard constraints and soft constraints. The former must not be violated under any circumstances, while some degree of latitude is permitted with the latter.

3.0 KNOWLEDGE AND CONSTRAINT BASED SCHEDULING

The existence of the soft (i.e, relaxable) constraints gives rise to inherent difficulties in the production scheduling problem. Without these, the problem could be attacked using reasonably conventional procedures

from the operations research domain. In many instances the soft constraints are dealt with by determining the relative importance and/or trade-offs among these constraints [26-28]. The fundamental challenge, therefore, is to formulate a software system that adequately captures the human expert intuition which is currently relied on for handling these constraints in the real world. This fundamental need to capture and implement human expertise suggests the applicability of the knowledge based system approach that has been undertaken in this study. Application of knowledge based techniques to anything, including scheduling, is a recent trend; some works specific to scheduling are reported in references 29-48. The results of an elaborate survey of various artificial intelligence based scheduling systems is presented in reference 49.

Associated with the development of the knowledge-based scheduling system given in this report was the concurrent evolution of a general model of a manufacturing facility. This model will constitute the conceptual base on which the methodology rests. The methodology takes into account the following elements of a production environment.

- resource characteristics such as processing capability, capacity and availability. Intuitively the resources correspond to the machines that carry out the processing activities but from a scheduling point of view, it is the set of available time intervals on these machines that is of primary importance.

- workpiece characteristics such as size and weight.

- job characteristics such as due date and quantity (number of parts). A job contains one or more identical parts treated as a unit, which requires a series of different manufacturing steps, usually manufactured with a specific customer in mind.

- process characteristics such as type and duration. The same machine may be used for different process types, possibly requiring different processing durations.

- various other constraints. Constraint representation, constraint relaxation, constraint modeling for the specific case and constraint application methodology along with the priority of application will form the core of the production scheduling system.

Some global properties of the target system can be expressed in terms of an input/output perspective. The system input will include a characterization of the current state of the production environment elements as given above. The system output will naturally include a schedule listing all jobs and the time and resources needed to complete the job. An indication of the degree to which any soft constraints are violated by the schedule would be a useful feature of the output. In the event that a feasible schedule encompassing all tasks is not possible, a knowledge-based solution procedure would be expected to provide some indication of the key impediments. Similar "advisory" output would also be expected for those situations where a feasible schedule is presented but its quality (as measured in terms of the degree to which soft constraints have been violated) is questionable.

4.0 MODEL DEVELOPMENT

A particular model for the scheduling problem has evolved during the course of this work. The model encompasses the principal objects with which the problem is concerned together with the interrelationships that exist among those objects. It provides a comprehensive framework to explore the development of solution procedures. It is, in fact, the basis for the knowledge-based solution procedures that have been developed in this study.

4.1 Model Parameters

The model consists of four main classes of interrelated objects derived from the manufacturing environment described earlier. Each object class is characterized by a set of attributes. The attribute set may vary from one production environment to another, depending on local priorities and corporate goals. The object classes are as follows.

(i) Resources

A resource is a generic service provider. A specific resource can provide a variety of service types. However, at any given time, only one type of service can be provided.

(ii) Time-Slots

A time-slot is a bounded interval of time that is "owned" by some resource. Each time-slot plays the role of an actual service provider. Among the attributes of a time-slot are the name of resource owning the slot, its left-hand boundary and its right-hand boundary. A resource may own many

time-slots, but they can never overlap each other. The time-slots are separated into three classes:

(a) scheduled-time-slot -- a time-slot during which the resource in question has been scheduled to perform a specific manufacturing process (and hence is not available for any other work);

(b) blocked-time-slot -- a time-slot during which the specific resource is not used for any manufacturing purpose and is tentatively unavailable for production, e.g., weekends, off shift hours, and preventive maintenance periods.

(c) free-time-slot -- a time-slot during which the associated resource is free and available for production.

(iii) Processes

A process is a generic service consumer. Each process has a requirement for a particular service type.

(iv) Tasks

Each task represents an actual service consumer and is regarded as being owned by a unique process. Task attributes include the job number under which the task is being performed and the type of process required by the task.

(v) Constraints

The problem-solving process is subject to a set of constraints that are stated in terms of the declared attributes of the resources, time-slots, processes and tasks which exist within the problem specification. These constraints may take a variety of forms and may include precedence, due date and physical constraints, as well as constraints resulting from company operational policies and corporate manufacturing goals associated with the facility.

4.2 Model Representation

The structural form of the model is shown in Figure 1. In this representation, the service requirement of each process is reflected by links to the various resources that can provide the service. The implied "many-to-many" relationship between the set of resources and the set of processes reflects the fact that the service requirement of each process can be provided by many resources and that each resource can provide the service requirement of several processes. Note also that the notion of a process can be viewed as a means for separating the tasks into equivalence classes that are based on the nature of the service

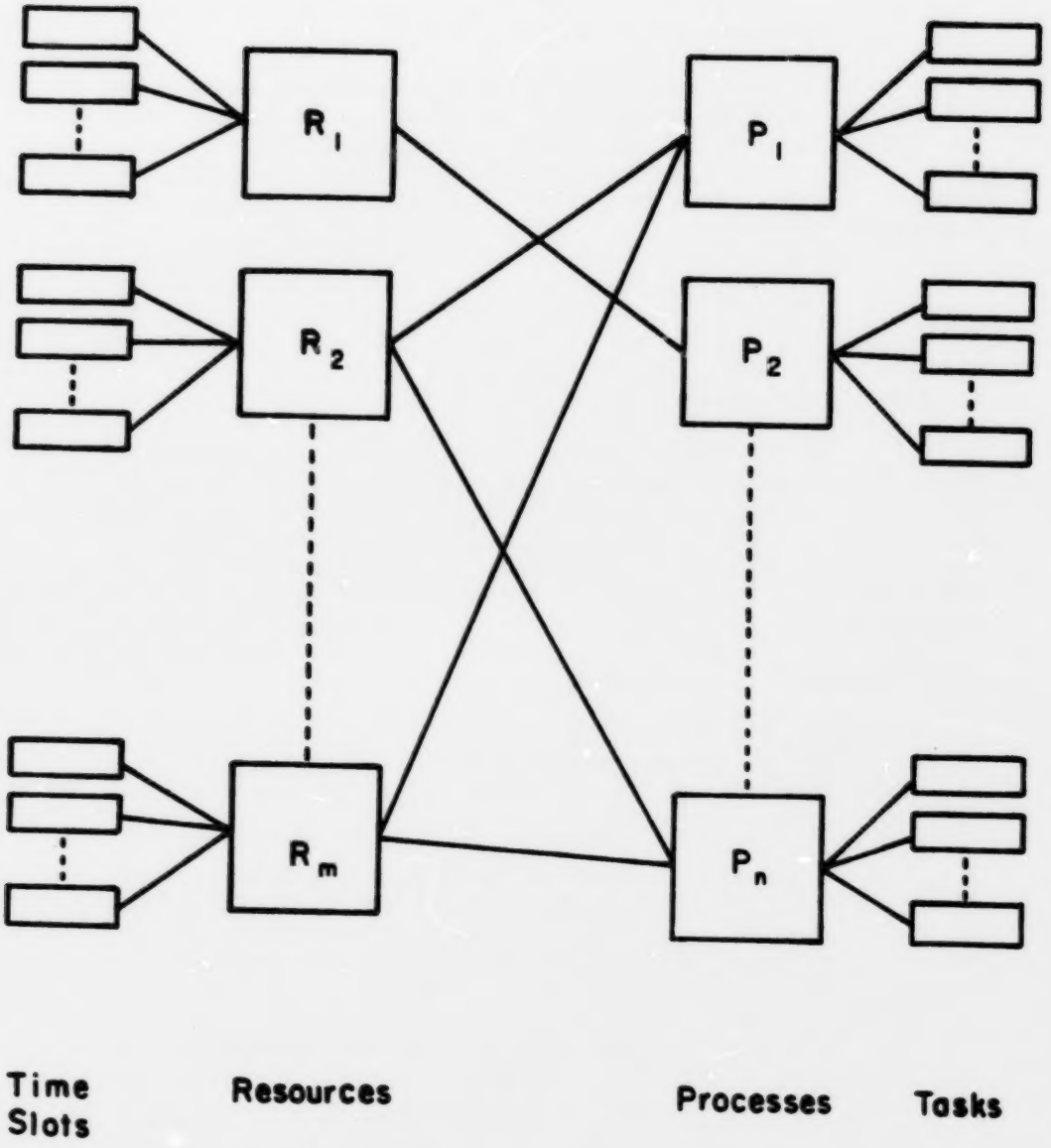

Figure 1: Structure of a Manufacturing Operation Model

requirement of the tasks. An analogous (but less intuitively appealing) interpretation can be given to the resource/ time-slot relationship.

In terms of the structure outlined above, the scheduling problem to be solved corresponds to mapping each of the tasks into a subinterval of some compatible time-slot (i.e., one to which it has a linkage) in such a way that the constraints in the constraint set are satisfied to some "acceptable" degree.

A noteworthy feature of the model is its structural symmetry. This, in particular, is consistent with the two ways that the problem can be regarded, namely, as a set of service providers looking for consumers or as a set of consumers looking for service providers.

It is interesting to note that the same model can be used to represent batch machine operations (where a number of parts together undergo a specific operation), job-shop operations (when appropriate precedence constraints are imposed), or flow-shop operations. In this report, the model described will be applied to a specific case i.e., a batch machine scheduling problem.

5.0 BATCH MACHINE SCHEDULING

The batch machine scheduling function is more complex than the general production scheduling function that is often referred to in the literature, for the following reasons.

(a) A batch is a group of tasks selected from a larger group of tasks which is awaiting identical processing. The batch, viewed as a distinct unit, must generally satisfy certain constraints with respect to the resource that will be used for processing it.

(b) At any given time, many batches may compete for the same resource. Scheduling, therefore, involves the selection of a specific batch instead of a specific task.

(c) In addition to the constraints mentioned in a) above, which arise from inherent characteristics of the available resources, the formation of a batch must also take into account the constraints associated with each of the constituent tasks, e.g., due date and earliest available time.

(d) The creation of a batch is normally based on some manufacturing strategy and may involve optimization criteria. In many instances these criteria and the associated data are available only in a qualitative form.

5.1 Batch Machine Problem Statement

A particular feature of the batch machine scheduling problem considered here is the absence of precedence constraints. The effect of these constraints is, however, implicitly accommodated by imposing due dates. Essentially, the problem elements are.

- n batch machines;
- m different processes;
- constraints associated with the machines, processes, tasks and company.

5.2 Batch Machine Constraints

An essential feature of the problem is the set of constraints that need to be taken into account in formulating a solution. Three types of constraints are incorporated in the problem formulation, namely, HARD, SOFT and PREFERENCE.

A PREFERENCE constraint frequently corresponds to the optimization (maximization/minimization) of some criterion, i.e., the achievement of a goal. This particular class of constraint, however, is distinctive because there is a wide latitude in the degree to which the goal's non-achievement can be tolerated. The distinction between SOFT and PREFERENCE constraints is not strictly defined but is maintained in this presentation for modeling convenience.

The specific constraints considered are the following.

- Physical constraints on machines --- HARD.

  Minimum volume, minimum weight -- the machine is alloved to operate only when these minimum conditions are met.

  Maximum volume, maximum weight -- the machine cannot operate beyond the limit imposed by the maxima.

- Process duration constraint --- HARD.

  A specific process carried out on a specific machine requires a fixed length of time.

- Earliest start time (EST) of a task -- HARD.

  The task will not be available earlier than a specific time; accordingly, the process cannot be started

earlier than this time.

- Latest start time (LST) of a task -- SOFT.

  If the task is not started by a certain time, it may result in a penalty.

- Capacity: working hours/day -- SOFT.

  Working hours per day are generally fixed; however, the management can extend this by scheduling overtime hours.

- Due date associated with each task -- SOFT.

  It is always desirable to meet the due dates; however, if there is not enough capacity, due date cannot be met.

- Company goals -- PREFERENCE.

  The company may have general operational policies such as minimization of work-in-process (WIPS) inventory, and setup time.

5.3 Batch Machine Solution Strategy

As mentioned earlier, with the batch machine scheduling problem, each resource can, in general, simultaneously accommodate a number of tasks that is consistent with the capabilities of the resource within a process type. However, constraints that are specific to the resource impose limits on this accommodation. The essential steps of the solution procedure (see Figure 3) are given below.

I. From among the currently available free time-slots, identify the "candidate time-slot" (cts), which is the one having the earliest start time. This implicitly identifies a particular resource R, namely, the one that "owns" the cts. Let Ts be the value of the left-hand boundary of the cts.

II. Identify the candidate processes that could be allocated to the cts. The conditions which must be satisfied by a candidate process are the following.

(a) The process must have a direct linkage to the resource R (see Figure 1).

(b) The processing time requirement must be less than (Tr - Ts), where Tr is the right-hand boundary of the cts (see Figure 2).

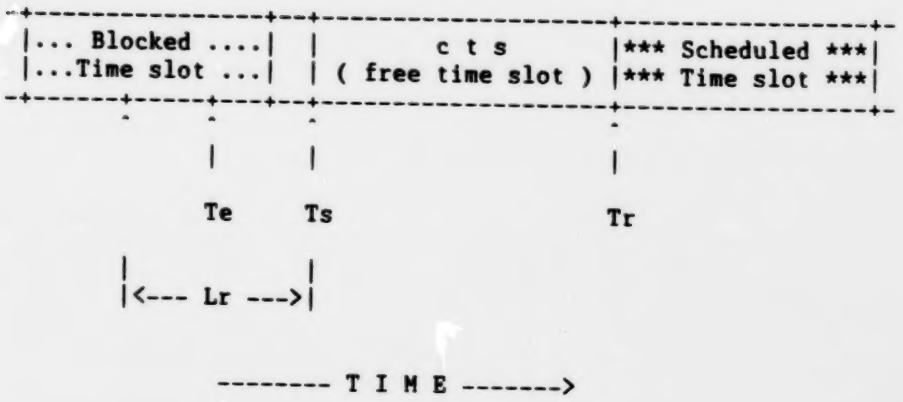

Figure 2: Time Slot Diagram

(c) Among the set of tasks owned by the process, there must exist a non-empty subset, S, that contains the tasks whose EST is less than or equal to Ts.

(d) The tasks in S must collectively satisfy some collection of lover bound constraints (e.g., minimum weight and/or minimum volume) associated with the resource R.

If at least one candidate process emerges, proceed to step IV, otherwise do step III.

III. Replace Ts with Ts' (where Ts' is the least value of time which is greater than Ts that coincides with the EST of some task which belongs to a process that is connected to the resource, R) and repeat until either

(a) at least one candidate process emerges or

(b) Ts exceeds the left-hand boundary of some other free-time-slot.

If (a) occurs, then proceed to step IV. If (b) occurs, tentatively block the current cts up to Ts, make the new free-time-slot the cts, set Ts to its left-hand boundary. and do step II.

IV. Batch Selection Procedure:

Each candidate process identified in steps II and III is associated with a set of tasks. An "optimal" subset of each such set is first determined and identified as a "candidate batch". From among these candidate batches, one is selected for tentative assignment to the cts.

The specification of each candidate batch is obtained as the solution of a (constrained) optimization subproblem. The constraints arise from the requirement that the tasks within a candidate batch must collectively satisfy some set of upperbound constraints associated with the resource R. The optimization is carried out with respect to a utility function that captures the "importance" of each task within the batch. The function is a linear combination of several terms that reflect a variety of relevant features; e.g., customer importance, management dictated urgency, value of the tasks, time to due date, time to LST.

The final batch selection is based on a prioritization of the various constraints. Let B be the set of candidate optimal batches established for the current cts and let u(b) denote the utility associated with batch b such that [b ε B]. Let b* denote the batch having the greatest utility (b* is assumed to be unique) and let c > 0 be a constant representing "utility indifference".

Determine B1 = {b ε B : [u (b*) - u (b)] < c}

(i.e., B1 represents the set of those batches that have a utility which is "close" to that of b*).

If B1 contains only b*, then assign b* to the cts.

Else,

Determine B2 = {b ε B1 : b is late}

Two types of lateness are recognized, namely due-date lateness and latest-start-time lateness. These occurrences are signaled by a negative value for the batch attributes due-status and lst-status respectively.

If B2 contains only one batch, assign it to the cts.

Else,

If B2 is empty, then make an assignment to the cts using the shortest processing time (SPT) rule over B1 (the SPT rule represents a manufacturing goal).

Else,

Determine B3 = {b ε B2 : b is "important"}.

A batch is considered important if at least one task within the batch is important. Task "importance" is a function of several factors, such as the customer rank, task urgency, and value of the task. When the value of this function is greater than a given constant, the task is considered important.

If B3 contains only one batch, assign it to the cts;

Else,

If B3 is empty, then make an assignment to the cts using the SPT rule over B2.

Else,

Determine B4 = {b ε B3 : b is LST late and due-date late}.

If B4 is not empty, then make an assignment to the cts using the SPT rule over B4.

Else,

Determine B5 = {b ε B3 : b is LST late}.

If B5 is not empty, then make an assignment to the cts using the SPT rule over B5.

Else,

Make an assignment to the cts using SPT rule over B3.

V. Scheduling procedure

Let C be the batch that results from step IV. Recall that C has met all HARD constraints and let

Lr = max { 0, -(lst-status), -(due-status) }
Te = max-est

In the above lst-status, due-status and max-est (maximum value of the EST's of the tasks within a batch) are all attributes of the batch element class of which C is an instance.

Recall (from Step II) that Ts is greater than or equal to Te. Within a free-time-slot the position of Ts can be (see Figure 2).

(i) Ts = the end time of a blocked-time slot (bts).

(ii) Ts = the end time of a scheduled-time-slot (sts) or Ts is within a free-time-slot. (This condition could happen if Ts was obtained by advancing to the right in the cts, to satisfy a lover bound physical constraint on the resource).

Case I. Ts = the end time of a bts.

If C meets all SOFT constraints (i.e., if Lr = 0), schedule it in the cts, starting at Ts.

Else,

If C has not met all SOFT constraints (i.e., Lr > 0) but it consists only of "unimportant" tasks, schedule it in the cts.

Else,

If C is "important" and if Te = Ts, then schedule the batch in the cts.

Else,

If Te < Ts and Lr > (Ts - Te), replace the value of Lr with (Ts - Te). (This condition can occur when there is a conflict between Te and one of the soft constraints LST or due date).

Or,

If Te < Ts and Lr $\leq$ (Ts - Te), do the following.

Overtime on resource R:- Create Tsn by relaxing Ts towards the left such that the Tsn is the largest value among

i) start of the scheduling period,
ii) Te, and
iii) immediate preceding scheduled-time-slot end time (if there is such a slot).

a) Let Trn = (Tsn + process duration)

If the time span between Trn and Tsn is available for scheduling, possibly with overtime, proceed to (b) below.

Else,

Modify Tsn by advancing it to the right to reach the time at which scheduling is allowed (with or without overtime), and return to (a) above.

b) If (Ts - Tsn) $\geq$ Lr, schedule C starting at Tsn;

Else,

Go to the section under 'overtime on any resource'.

Case II. Ts $\neq$ the end time of a bts.

If Ts $\neq$ the end time of a bts, create Tsn = Ts and do the following.

Overtime on any Resource:- Disqualify those resources whose physical constraints cannot accommodate the batch C. Create Tsnz such that it is the greater of:

i) start of scheduling period and
ii) Te.

(a) Initialize Tsnx = Tsnz, where x corresponds to each remaining resource.

(b) Set Tsny = min {Tsnx over all valid x}.

(c) Calculate Trn = (Tsny + duration on resource y).

If the time span between Trn and Tsny is available for sheduling proceed to d) below.

Else,

Modify Tsnx of the resource y by advancing it to the right to reach the time at which scheduling is allowed (with or without overtime) and return to (b) above.

(d) If Tsny < Tsn schedule the batch starting at Tsny.

Else,

Go back and schedule the batch at the original cts starting at Tsn.

VI. Appropriately update the set of free-time-slots and the set of tasks to reflect the scheduling carried out in step V and then return to step I.

A summary of the solution procedure, described above, is provided in Figure 3.

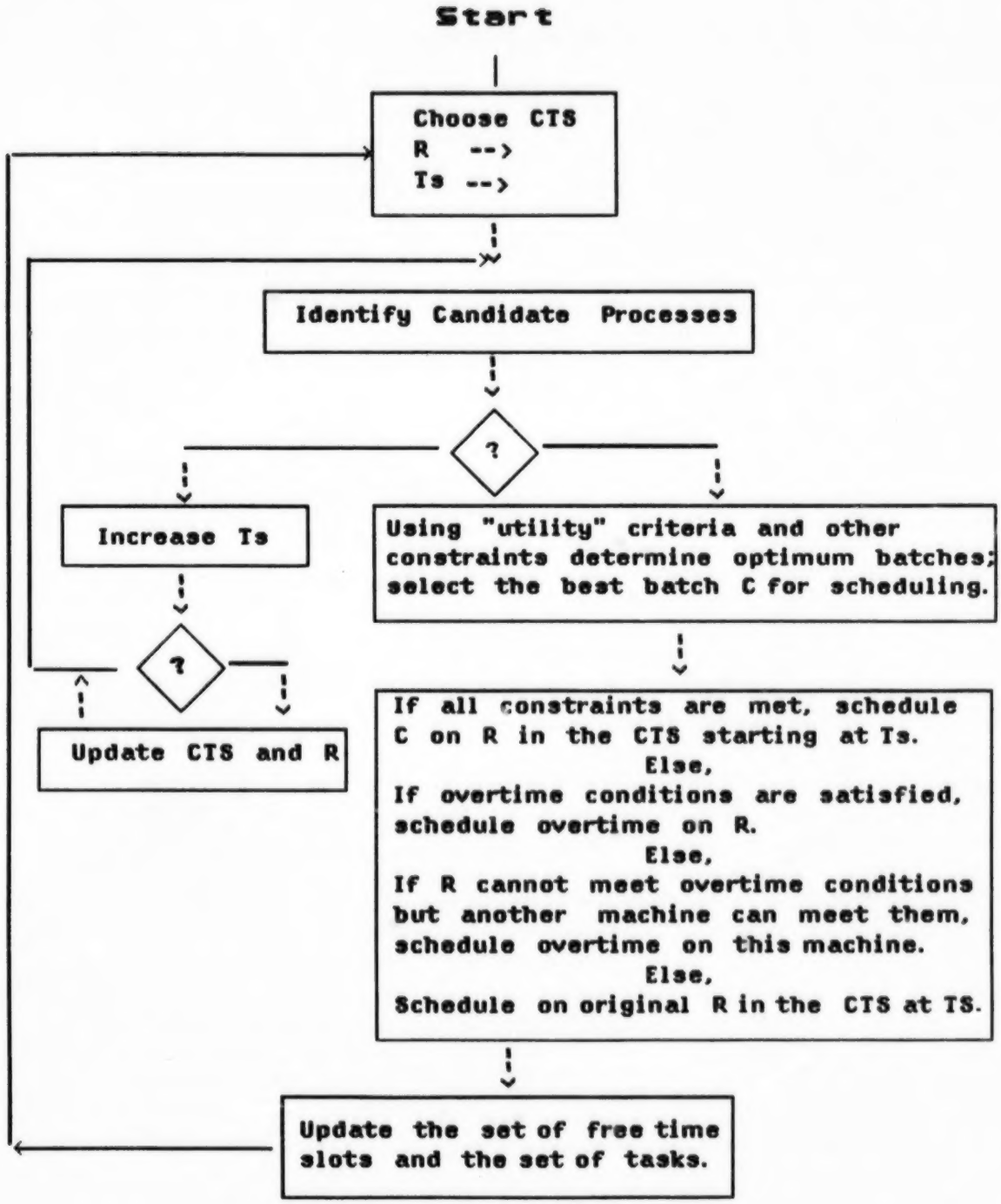
Start
Choose CTS
R -->
Ts -->
Identify Candidate Processes
?
Increase Ts
Using "utility" criteria and other constraints determine optimum batches; select the best batch C for scheduling.
?
Update CTS and R
If all constraints are met, schedule C on R in the CTS starting at Ts.
Else,
If overtime conditions are satisfied, schedule overtime on R.
Else,
If R cannot meet overtime conditions but another machine can meet them, schedule overtime on this machine.
Else,
Schedule on original R in the CTS at TS.
Update the set of free time slots and the set of tasks.

Figure 3: Batch Machine Solution Procedure

6.0 PROGRAM ORGANIZATION AND DEVELOPMENT

The program code that was developed to support the conceptual solution approach outlined in the previous sections is organized into two modules. The first of these encompasses the set of rules upon which the underlying knowledge- and constraint-based scheduling concept is based. This module has been implemented in the OPS5 expert system development environment. The second module is a collection of FORTRAN subprograms that provide a support function for the OPS5 module. Each of these modules is briefly outlined below. In general, functions that are algorithmic in nature are handled by a series of FORTRAN subprograms, whereas functions that involve logical decision making are dealt with by a set of OPS5 rules.

6.1 The OPS5 Module

The rules contained in this module can be separated in five distinct groups that correspond to various functional tasks within the solution procedure. These are

(a) generation of the initial problem state from data maintained in external data files,
(b) selection of a candidate time slot,
(c) selection of candidate processes,
(d) allocation of an optimal batch to a candidate time slot, and
(e) scheduling of overtime.

Movement from one of these "rule-sets" to another is controlled by the use of an element class called "controller" which has an attribute called "state". Only one instance of this element class exists within the OPS5 program and a condition element of most rules makes reference to it. Movement of program activity from one rule set (functional task) to another is then achieved by appropriately setting the value of the state attribute of the controller entity.

Problem initialization is primarily concerned with the creation of instances of the various element classes that correspond to the (available) time-slots, resources, processes, and tasks. These characterize the specific scheduling problem that is to be treated. For this the FORTRAN module is utilized and some further details are provided in the following section 6.2.

The identification of each successive candidate time-slot (cts) is a principal functional step in the solution procedure. The rules involved in this activity carry out steps I, II, and III that are outlined in section 5.3. The identification of the candidate processes associated with a cts is a step that is closely related to the cts selection

procedure itself. Thus the rule-sets (b) and (c) are tightly inter-related.

In general, several processes can be accommodated by a particular time-slot, and for each such candidate process, an optimal batch is created (using the services of the FORTRAN module as described in section 6.2). It remains, therefore, to select one such optimal batch for actual allocation to the cts. The considerations involved in this selection are outlined in step IV of section 5.3 and the rules in the rule-set (d) carry out this selection procedure.

If scheduled in the cts, the optimal batch identified by rule-set (d) may violate the soft lateness constraints associated with latest start time and/or due date. In such a case, the solution procedure undertakes to better accommodate the selected optimal batch by scheduling overtime, as described in step V of section 5.3. This could occur on either the resource associated with the cts or alternately on some other resource. The rules in rule-set (e) implement the strategy for carrying out this phase of the solution procedure.

6.2 The FORTRAN Module

Separate portions of the FORTRAN module participate in three particular phases of the underlying solution procedure. These are

(a) problem initialization,
(b) determination of an optimal batch for each active operation, and
(c) removal from further consideration, of tasks that have been scheduled.

In each of these cases, the required service of the FORTRAN module is initiated by appropriate subprogram references from the right-hand side of one or more OPS5 rules.

Problem initialization is concerned with the creation of instances of various OPS5 element classes that, at least in part, provide the specifications for the scheduling problem within the OPS5 framework. These instances are created from data contained in three separately maintained data files, which respectively contain specifications for

(i) The "blocked" time-slots on the various resources associated with the problem,
(ii) The (previously) scheduled time-slots on the various resources associated with the problem, and
(iii) The order data associated with the problem (these give rise to the tasks that need to be scheduled).

Each time a cts is identified in the basic procedure, it is necessary to establish an optimal batch for each of the processes that could be allocated to that cts (or rather to the resource that owns the cts). This is essentially an algorithmic procedure, which is best carried out within the FORTRAN environment. The following steps are involved: transfer to the FORTRAN environment of the task identifiers from which an optimal batch (for some particular process) will be constructed, determination of the optimal batch, and finally, creation (within the OPS5 environment) of an instance of the batch element class that holds the various pertinent specifications of the optimal batch.

A particular subset of OPS5 rules determines which of the various optimal batches will be selected for allocation to the candidate time-slot. Once this is achieved, the tasks within this particular batch are then clearly "accommodated" and must, therefore, be removed from the set of tasks that remain to be scheduled. This removal activity relies on the services of a portion of the FORTRAN module.

7.0 TEST PROBLEM DESCRIPTION

The data that specifies the pertinent features of an example problem is outlined in this section. The scheduling period is taken to extend between time = 48 and time = 200. Three processes called A, B, and C are considered and it is assumed that each can be accommodated on the two resources (called Furnace1 and Furnace2) which are associated with the problem. The duration of each process when carried out on the two resources is summarized below.

Process Name	Resource Name	Duration
A	Furnace1	8
A	Furnace2	10
B	Furnace1	12
B	Furnace2	15
C	Furnace1	20
C	Furnace2	25

Both resources, Furnace1 and Furnace2, are associated with "physical" (HARD) constraints which are interpreted as weight and volume constraints. These are as follows:

Furnace1:	min. weight	10	max. weight	40
	min. volume	5	max. volume	20
Furnace2:	min. weight	11	max. weight	41
	min. volume	5	max. volume	21

Within the scheduling interval (i.e., [48, 200]) both resources have time periods that are unavailable for scheduling. These are as follows (see also Figure 4):

Furnace1: previously scheduled period: [95,110]
blocked periods: [50, 60]; [150, 200]

Furnace2: blocked period: [55, 75]
previously scheduled periods: [98, 125]; [175,190]

A summary of the customer order data (from which the tasks to be scheduled are derived) is given in Table 1.

job	proc-name	due-date	est	lst	qty	unit-vol	unit-wgt	rank	lot-value	urgency
111	B	176	46	76	1	3	8	75	40	0
112	B	80	47	52	1	3	50	100	50	0
113	B	80	36	73	2	7	13	100	30	0
114	B	32	46	55	2	5	15	75	40	0
115	B	104	50	78	1	6	21	75	90	0
116	B	152	44	61	1	1	12	75	60	100
221	A	104	49	76	1	6	11	100	30	0
222	A	56	39	49	2	8	16	75	70	0
223	A	54	42	74	1	7	7	100	10	0
224	A	104	48	61	1	8	10	100	70	0
225	A	128	45	60	2	4	8	100	50	0
331	C	90	63	76	2	3	8	75	40	0
332	C	176	56	86	1	2	10	100	50	0
333	C	152	44	66	1	4	5	75	30	0
334	C	104	40	69	3	5	9	100	55	0
335	C	176	56	86	1	24	10	100	50	0

job = job number
proc-name = name of the required process
due-date = required completion time
est = earliest start time
lst = latest start time
qty = number of pieces produced under this job
unit-vol = unit volume
unit-wgt = unit weight
rank = customer rank
lot-value = value of the lot
urgency = management dictated priority

Table 1: Customer Order Data:

Note: In this data, the due-date and the lst are unrelated.

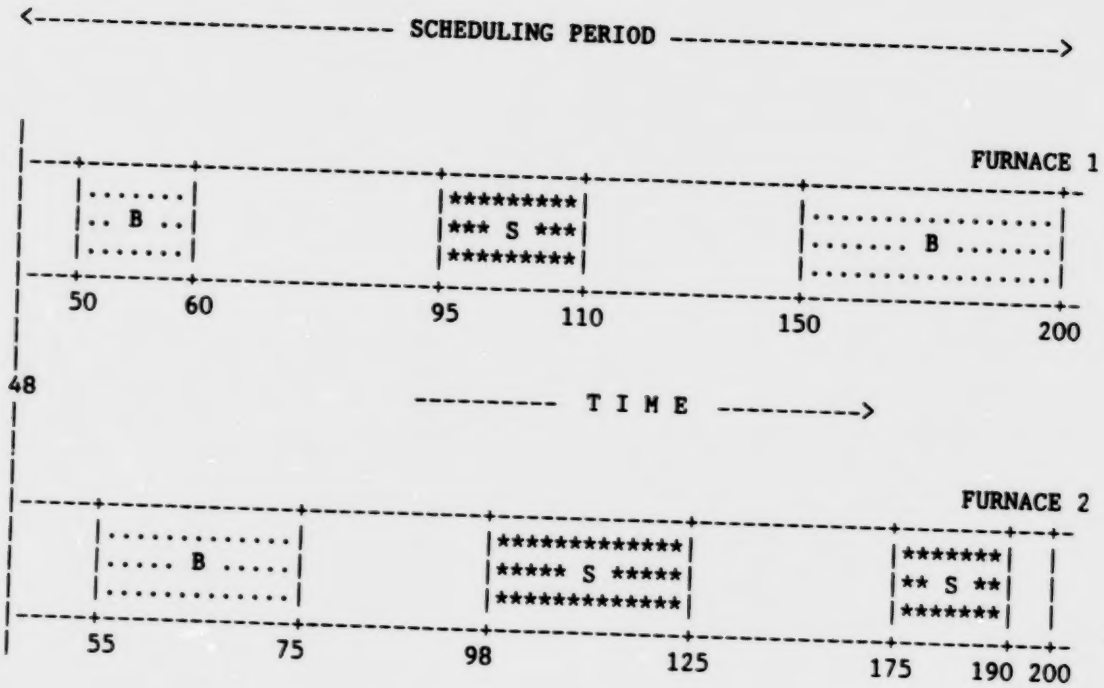

FIGURE 4: GANTT CHART (Initial)

8.0 DISCUSSION OF RESULTS

The software system described in this report has been designed to schedule all available tasks on appropriate machines within the constraints of the scheduling period (planning horizon). For the example problem described in section 7, Figure 5 shows the Gantt chart when the blocked-times corresponding to all machines are available for over-time whereas Figure 6 shows the Gantt chart when overtime is not available on any machine. Similarly, Table 2 depicts the schedules for each task for the cases with and without overtime. The Table 2 also points out that tasks 112 and 335 cannot be scheduled because they violate some physical constraints.

It can be noticed from Table 3 that the average lateness per task was halved when overtime was imposed on all resources. However, this is somewhat insignificant. The important aspects (which are not apparent from Table 3 and from Figures 5 and 6) are primarily qualitative factors such as customer satisfaction arising from early delivery of the product, meeting of due dates, reduction of penalties and delay costs.

9.0 CONCLUSIONS

KCBS software has been developed and is being used as an experimental scheduling system. It can schedule all available tasks within the constraints of the scheduling horizon. Most importantly, it automatically schedules overtime if such overtime is permissible and if the tasks being scheduled are "important" enough.

The system is compossed of over 100 OPS5 rules and a number of FORTRAN routines and is expected to grow. At present, the model and the methodology developed are used as learning tools. However, the experience acquired from this exercise will be directly useful in the joint project between Systems Laboratory, NRC and Huyck Canada of Arnprior, Ontario, which has been initiated already.

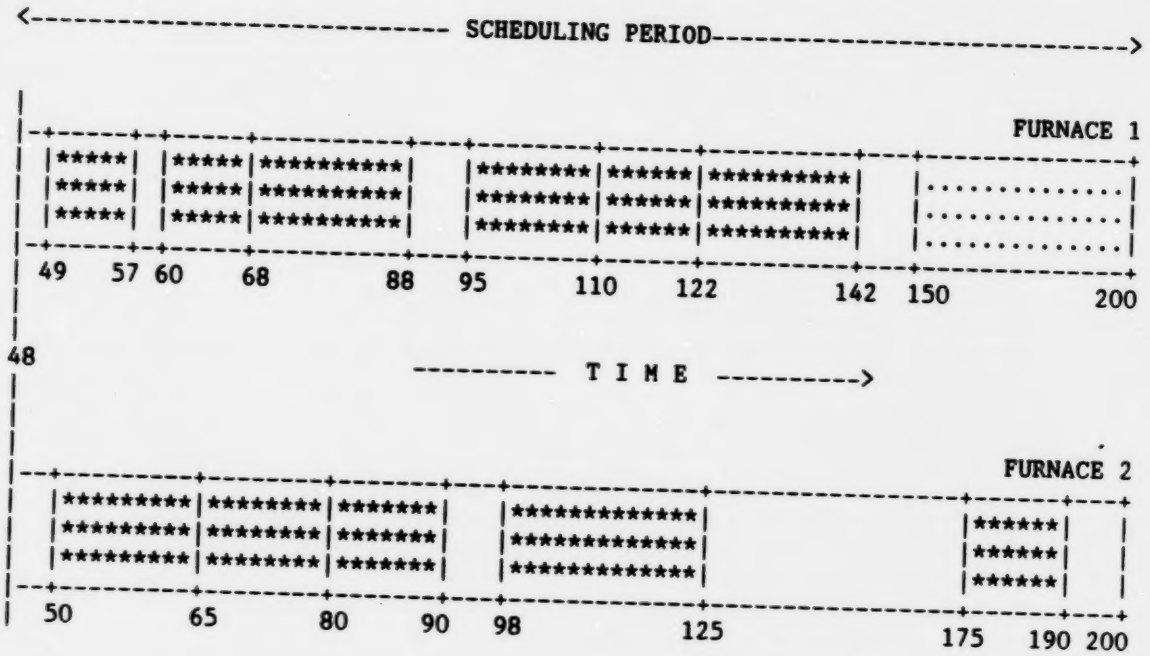

FIGURE 5: GANTT CHART (With Overtime)

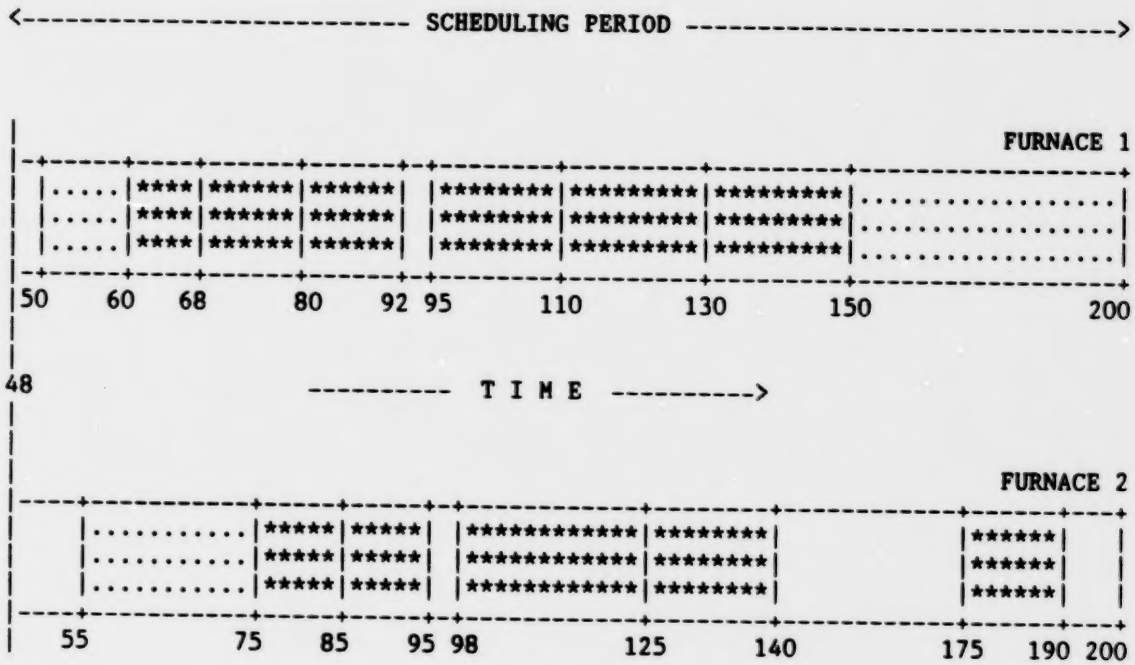

FIGURE 6: GANTT CHART (Without Overtime)

job	due-date	lst	unit-vol	unit-wgt	Schedule without overtime start	Schedule without overtime end	Schedule with overtime start	Schedule with overtime end
111	176	76	3	8	80	92	65	80
113	80	73	7	13	80	92	110	122
114	32	55	5	15	125	140	65	80
115	104	78	6	21	68	80	50	65
116	152	61	1	12	68	80	50	65
112	80	52	3	50	invalid task -- check weight constraint			
221	104	76	6	11	75	85	49	57
222	56	49	8	16	85	95	80	90
223	54	74	7	7	75	85	49	57
224	104	61	8	10	60	68	60	68
225	128	60	4	8	60	68	60	68
331	90	76	3	8	110	130	68	88
332	176	86	2	10	110	130	68	88
333	152	66	4	5	110	130	68	88
334	104	69	5	9	130	150	122	142
335	176	86	24	10	invalid task -- check volume constraint			

Table 2: Schedule Comparison -- Overtime vs No overtime

job	due-date	lst	Lateness without overtime due-date-late	Lateness without overtime lst-late	Lateness with overtime due-date-late	Lateness with overtime lst-late
111	176	76	0	-4	0	0
113	80	73	-12	-7	-42	-37
114	32	55	-108	-70	-48	-10
115	104	78	0	0	0	0
116	152	61	0	-7	0	0
221	104	76	0	0	0	0
222	56	49	-39	-36	-34	-31
223	54	74	-31	-1	-3	0
224	104	61	0	0	0	0
225	128	60	0	0	0	0
331	90	76	-40	-34	0	0
332	176	86	0	-24	0	0
333	152	66	0	-44	0	-2
334	104	69	-46	-61	-38	-53
Average Lateness:			-19.7	-20.57	-11.7	-9.5

Table 3: Lateness Comparison: Overtime vs No Overtime

10.0 REFERENCES

1. BUCKLEY, J., CHAN, A., GRAEFE, U., NEELAMKAVIL, J., SERRER, M., and THOMSON, V. 1987. An Integrated Production Planning and Scheduling System for Manufacturing Plants. MSTF'87. International Conference on the Manufacturing Science and Technology of the Future, Boston, MA.

2. BIRTA, L. G. 1987. An Investigation of Some Knowledge-Based Issues Relating to Production Scheduling Problems Using OPS5. Final Report. NRC Contract # 042-1738/3678Y, Canada.

3. FOX, M.S. 1983. Constraint-Directed Search: A Case Study for Job Shop Scheduling. Ph.D Thesis, Carnegie-Mellon University, Pittsburg, MA.

4. FOX, M.S., and SMITH, S.F. 1984. ISIS - A Knowledge Based System for Factory Scheduling. Expert Systems, 1, (1): pp: 25-49.

5. FOX, M.S., and ALLEN, B.P. 1983. ISIS: A Constraint-Directed Reasoning Approach to Job Shop Scheduling. Carnegie-Mellon University Technical Report CMU-RI-TR-83-8.

6. FOX, M.S., and SMITH, S.F. 1984. Constraint-Based Scheduling in an Intelligent Logistics Support System: An Artificial Intelligence Approach. Carnegie-Mellon University. Annual Report, AFOSR Contract No. F49620-82-K-0017.

7. FOX, M.S., and McDERMOTT, J. 1986. The Role of Databases in Knowledge-Based Systems. Carnegie-Mellon University Technical Report CMU-RI-TR-86-3.

8. FOX, M.S. 1986. Observations on the Role of Constraints in Problem Solving. Canadian AI Conference, Montreal, Quebec.

9. SMITH, S.F., FOX, M.S., and OW, P.S. 1986. Constructing and Maintaining Detailed Production Plans: Investigations into the Development of Knowledge-Based Factory Scheduling Systems. AI Magazine 7 (4) pp: 45-61.

10. SMITH, S.F., and OW, P.S. 1985. The Use of Multiple Problem Decompositions in Time Constrained Planning Tasks. Proceedings of the 9th IJCAI, Los Angeles, CA.

11. SMITH, S.F. 1983. Exploiting Temporal Knowledge to Organize Constraints. Carnegie-Mellon University Technical Report CMU-RI-TR-83-12.

12. STEFIK, M. 1981. Planning with Constraints - MOLGEN: Part 1. Artificial Intelligence 16: pp: 111-140.

13. DECHTER, R., and PEARL, J. 1985. The Anatomy of Easy Problem: A Constraint Satisfaction Formulation. Proceedings of the 9th IJCAI, Los Angeles, CA.

14. FREUDER, E.C., and QUINN, M.J. 1985. Taking Advantage of Stable Sets of Variables in Constraint Satisfaction Problems. Proceedings of the 9th IJCAI, Los Angeles, CA.

15. STICKEL, M.E., and TYSON, W.M. 1985. An Analysis of Consecutively Bounded Depth-First Search with Applications in Automated Deduction. Proceedings of the 9th IJCAI, Los Angeles, CA.

16. MOTTELER, H.E., and KANAL, L.N. 1985. The Complexity of Searching Several Classes and/or Graphs. Proceedings of the 9th IJCAI, Los Angeles, CA.

17. ERSCHLER, R., ROUBELLAT, F., and VERNHES J.P. 1980. Characterizing the Set of Feasible Sequences for n Jobs to be carried out on a Single machine. European Journal of Operational Research 4: pp. 189-194.

18. DECHTER, R., and PEARL, J. 1987. Network-Based Heuristics for Constraint-Satisfaction Problems. Artificial Intelligence, 34, (1): pp. 1-38.

19. CHANDRA, N., and MARKS, D. H. 1986. Intelligent Use of Constraints for Activity Scheduling. Applications of AI in Engineering Problems - First International Conference. Springer-Verlag, New York, NY.

20. HARALICK, R.M., and ELLIOT G.L. 1980. Increasing Tree Search Efficiency for Constraint Satisfaction Problems. Artificial Intelligence 14: pp: 263-313.

21. SMITH, S. F. 1987. A Constraint-Based Framevork for Reactive Management of Factory Schedules. Proceedings of the International Conference on Expert Systems and the Leading Edge in Production Planning and Control, Charleston, SC.

22. LE PAPE, C., and SMITH, S. F. 1987. Management of Temporal Constraints for Factory Scheduling. Carnegie-Mellon University Technical Report: CMU-RI-TR-87-13, USA.

23. STEFFEN, M.S., and GREENE, T. J. 1986. Hierarchies of Sub Periods in Constraint-directed Scheduling. Symposium on Real Time Optimization in Automated Manufacturing Facilities, National Bureau of Standards, Gaithersburg, MD.

24. MILNE, R. 1986. Constraint Driven Distribution Scheduling. Applications of AI in Engineering Problems - First International Conference, Springer-Verlag, Nev York, NY.

25. BERLINER, H., and GOETSCH, G. 1985. A Study of Search Methods: The Effect of Constraint Satisfaction and Adventurousness. Proceedings of the 9th IJCAI, Los Angeles, CA.

26. DESCOTTE, Y., and LATOMBE, J.C. 1985. Making Compromises among Antagonist Constraints in a Planner. Artificial Intelligence, 27: pp: 183-217.

27. WORRALL, B.M. 1986. Job Shop Scheduling Under Resource Constraints and Varying Order Priorities. Proceedings of the 8th Annual Conference on Computers and Industrial Engineering, Orlando, Fl.

28. WHITE, C.C, SAGE, A.P., and DOZONO, S. 1984. A model of Multiattribute Decision Making and Trade-off Weight Determination under Uncertainty. IEEE Transactions on Systems, Man and Cybernetics, SMC-14, (2): pp: 223-229.

29. SHEN, S., and CHANG, Y.L. 1986. An AI Approach to Schedule Generation in A Flexible Manufacturing System. Proceedings of the 2nd ORSA/TIMS Conference on Flexible Manufacturing Systems, Amsterdam, The Netherlands.

30. TATE, A. 1985. A Review of Knowledge-Based Planning Techniques. Expert Systems 85. Proceedings of the 5th Technical Conference of the British Computer Society Specialist Group on Expert Systems, University of Warwick, England.

31. LE PAPE, C. 1985. SOJA: A Daily Workshop Scheduling System. Expert Systems 85. Proceedings of the 5th Technical Conference of the British Computer Society Specialist Group on Expert Systems, University of Warwick, England.

32. BENSANA, E., CORREGE, M., BEL, G., and DUBOIS, D. 1986. An Expert System Approach to Industrial Job-shop Scheduling. IEEE International Conference on Robotics and Automation, San Francisco, CA.

33. GOLDSTEIN, I.P., and ROBERT, R.B. 1977. NUDGE: A Knowledge-Based Scheduling Program. MIT Publications, AI Memo. No. 405.

34. BRUNO, G., ELIA, A., and LAFACE, P. 1986. A Rule-Based System to Schedule Production. IEEE Computer, 19, (7) pp: 32-40.

35. FUKUMORI, K. 1980. Fundamental Scheme for Train Scheduling. MIT Publications, AI Memo. No. 596.

36. WILKINS, D.E. 1981. Parallelism in Planning and Problem Solving: Reasoning About Resources. SRI Project 8871, Technical Note 258.

37. ZEMEL, R., and ACOCK, M. 1986. DISPATCHER: An Intelligent Approach to Factory Control. IEEE American Control Conference, Seattle, WA.

38. SEMECO, A.C., WILLIAMS, B.D., ROTH, S., and GILMORE, J.F. 1986. GENSCHED - A Real World Hierarchical Planning Knowledge Based System. Proceedings of the SPIE Intnternational Society for Optical Engineering; Applications of Artificial Intelligence III, Orlando, Florida.

39. ERSCHLER, J., and ESQUIROL, P. 1986. Decision-Aid in Job Shop Scheduling: A Knowledge-Based Approach. IEEE International Conference on Robotics and Automation, San Francisco, CA.

40. ROBBINS, J.H. 1985. PEPS: The Prototype Expert Priority Scheduler. Autofact '85 November, Detroit, MI.

41. CHIODINI, V. 1986. A Knowledge Based System for Dynamic Manufacturing Replanning. Proceedings of a Symposium held at the National Bureau of Standards, Gaithersburg, MD.

42. LAWRENCE, S.R., and MORTON, T.E. 1986. PATRIARCH: Hierarchical Production Scheduling. Real-Time Optimization in Automated Manufacturing Facilities, Proceedings of a Symposium (NBS-SP-724). Gaithersburg, MD.

43. OW, P.S. 1986. Experiments in Knowledge-Based Scheduling. Carnegie-Mellon University Research Report: Contract No. F49620-82-K-0017.

44. ORCIUCH, E., and FROST, J. 1984. ISA: Intelligent Scheduling Assistant. First Conference of Artificial Intelligence Applications, IEEE Computer Society, Denver, Co.

45. KANET, J.J., and ADELSBERGER, H.H. 1987. Expert Systems in Production Scheduling. European Journal of Operations Research 29: pp: 51-59.

46. ALEXANDER, S.M. 1987. An Expert System for the Selection of Scheduling Rules in a Job Shop. Computers ind Engineering, 12, (3): pp: 167-171.

47. CHRYSSOLOURIS, G., WRIGHT, K., PIERCE, J., and COBB, W. 1987. Manufacturing Systems Operation: Dispatch Rules Versus Intelligent Control. MSTF'87 International Conference on the Manufacturing Science and Technology of the Future, Boston, MA.

48. CHOI, J. U., and BYRD, T. 1987. Blackboard Model Implementation in a Knovledge-Based Job-Shop Scheduling System. 2nd International Conference on Computers and Applications, Beijing, China.

49. STEFFEN, M.S. 1986. A Survey of Artificial Intelligence Based Scheduling Systems. Fall Industrial Engineering Conference, Boston, MA.

REPORT DOCUMENTATION PAGE/PAGE DE DOCUMENTATION DE RAPPORT

REPORT/RAPPORT TR-SYS-017 1a	REPORT/RAPPORT NRC No. 28863 1b
SECURITY CLASSIFICATION/ CLASSIFICATION DE SÉCURITÉ Unclassified 2	DISTRIBUTION/DIFFUSION ☐ Controlled/Contrôlé ☑ Unlimited/Illimité 3

TITLE/SUBTITLE/TITRE/SOUS-TITRE

KNOWLEDGE AND CONSTRAINT BASED SCHEDULING (KCBS) USING OPS5

4

AUTHOR(S)/AUTEUR(S)

Joseph Neelamkavil, Louis G. Birta

5

SERIES/SÉRIE

6

CORPORATE AUTHOR/PERFORMING AGENCY/AUTEUR D'ENTREPRISE/AGENCE D'EXÉCUTION

7

SPONSORING AGENCY/AGENCE DE SUBVENTION

8

DATE	FILE/DOSSIER	LAB. ORDER COMMANDE DU LAB.	PAGES	DIAGS	REFS
1988/04 9	10	11	30 12a	6 12b	49 12c

NOTES

13

DESCRIPTORS(KEY WORDS)/MOTS-CLÉS

Production Scheduling, Knowledge-Based, Rule-Based, Artificial Intelligence, Constraints, Scheduling System, Expert System

14

SUMMARY/SOMMAIRE

Describes a knowledge-based scheduling system and its associated scheduling methodology developed using OPS5 expert system shell and FORTRAN support routines.

15

ADDRESS/ADRESSE

Dr. Vince Thomson
Head, Systems Laboratory Tel: (613) 993-9461
Division of Mechanical Engineering
National Research Council Canada
Montreal Road, Ottawa, Ontario K1A 0R6

16